ON THE WAVELET OPTIMIZED FINITE DIFFERENCE METHOD

Leland Jameson[1]

Institute for Computer Applications in Science and Engineering

NASA Langley Research Center

Hampton, VA 23681

ABSTRACT

When one considers the effect in the physical space, Daubechies-based wavelet methods are equivalent to finite difference methods with grid refinement in regions of the domain where small scale structure exists. Adding a wavelet basis function at a given scale and location where one has a correspondingly large wavelet coefficient is, essentially, equivalent to adding a grid point, or two, at the same location and at a grid density which corresponds to the wavelet scale. This paper introduces a wavelet-optimized finite difference method which is equivalent to a wavelet method in its multiresolution approach but which does not suffer from difficulties with nonlinear terms and boundary conditions, since all calculations are done in the physical space. With this method one can obtain an arbitrarily good approximation to a conservative difference method for solving nonlinear conservation laws.

[1]This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681. Research was also supported by AFOSR grant 93-1-0090, by DARPA grant N00014-91-4016, and by NSF grant DMS-9211820.

1 Introduction

In the numerical simulation of equations which model physics it is common that small scale structure will exist in only a small percentage of the domain. If one chooses a uniform numerical grid fine enough to resolve the small scale features then in the majority of the domain the solution to the equations will be over resolved. One would like, ideally, to have a dense grid where small scale structure exists and a sparse grid where the solution is composed only of large scale features.

Consider now a Daubechies-based wavelet system. Wavelets provide a natural mechanism for decomposing a solution into a set of coefficients which depend on scale and location. One can then work with the solution in a compressed form where one works only with the wavelet coefficients which are larger in magnitude than a given threshold. Wavelets, therefore, sound ideal for solving the type of problem mentioned the previous paragraph. There are, however, serious problems matching the order of differentiation accuracy at the boundary for nonperiodic boundary conditions, see [9], with the superconvergence encountered with periodic boundary conditions, see [7]. Furthermore, wavelet methods generally require a tranformation between the physical space and the coefficient space for either evaluation of nonlinear terms or for differentiation.

In this paper a wavelet method which satisfies the goals of the first paragraph while using the wavelet machinery outlined in the second paragraph without the accompanying difficulties encountered at the boundaries and the expense of constantly tranforming between the physical space and the coefficient space will be introduced. That is, the new method utilizes the strength of wavelets, scale detection and data compression, while avoiding the difficulties by using wavelets in their finite difference form.

The following is a list of the sections of this paper with the noteworthy points.

1. **Introduction**

2. **Wavelet Definitions and Notation**

3. **Finite Difference Grid Refinement and Wavelets:**

   This section will establish that Daubechies-based wavelet methods are equal to finite-difference methods with grid refinement.

4. **The Wavelet-Optimized, Adaptive Grid, Finite Difference Method:**

   In this section a new numerical method which utilizes the strength of wavelets and avoids the difficulties will be proposed. That is, wavelets will be utilized to define the grid for finite difference methods. The new method is named, 'The Wavelet-Optimized, Adaptive Grid, Finite Difference Method', or simply 'WOFD'.

5. **WOFD applied to Burgers' equation:**

   Numerical results of the wavelet-optimized, adaptive grid, finite difference method applied to Burgers' equation with periodic and nonperiodic boundary conditions will be given.

6. **Accuracy of WOFD:**

   The error in finite-difference derivative approximation on a 5-point stencil is of the form,

$$Err = \Delta_1\Delta_2\Delta_3\Delta_4\frac{1}{120}f^{(v)}(a).$$

   Think of $f(x)$ as a pure mode, $f(x) = e^{ix\xi}$, where ξ is frequency or wave number. When the data is locally smooth, i.e., composed of low frequencies, the wavelet coefficients are small and consequently the Δ's are allowed to be large. When the data is locally oscillatory, i.e., composed of high frequencies, the wavelet coefficients are large and WOFD reduces the size of the Δ's. The effect is that the derivative approximation error will not grow faster than linearly with

respect to frequency. Recall that without grid adjustment this error would grow as a fifth power of frequency for a fourth order scheme.

7. **Stability of WOFD:**

   Analytical stability methods are beyond reach due the arbitrary nature of the grid. But, in practice the method displays no instability when applied to Burgers' equation.

8. **Efficiency of WOFD:**

   The WOFD method finds an approximation to the solution found on the finest scale across the whole domain. The efficiency depends on the rate of data compression. That is, if the finest scale has N grid points and the WOFD averages, say, N_0 grid points, then the WOFD method will find the solution using, roughly, $\frac{N_0}{N}$ times the number of operations used to find the finest grid solution.

9. **WOFD in Higher Dimensions**

   The discussion here will be limited to grid selection. It will be seen from a few examples that WOFD is an effective method for grid selection in higher dimensions. The examples given are for two dimensions. The local 'spectral analysis' of a wavelet method provides exactly the information needed to thoroughly understand the data and, hence, define a grid properly.

10. **Conclusion:**

    The WOFD method is an efficient and stable alternative to a Daubechies wavelet method. The WOFD method and a wavelet method are essentially the same. The only significant difference is the manner in which the grid is refined. The WOFD method, by contrast, avoids difficulties with nonlinear terms and boundaries by performing all calculations in the physical space.

2 Wavelet Definitions and Relations

The term wavelet is used to describe a spatially localized function. 'Localized' means that the wavelet has compact support or that the wavelet almost has compact support in the sense that outside of some interval the amplitude of the wavelet decays exponentially. We will consider only wavelets that have compact support and that are of the type defined by Daubechies [4] which are supported on $[0, 2M-1]$, where M is the number of vanishing moments defined later in this section.

To define Daubechies wavelets, consider the two functions $\phi(x)$ and $\psi(x)$ which are solutions to the following equations:

$$\phi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k \phi(2x-k), \tag{1}$$

$$\psi(x) = \sqrt{2}\sum_{k=0}^{L-1} g_k \phi(2x-k), \tag{2}$$

where $\phi(x)$ is normalized,

$$\int_{-\infty}^{\infty} \phi(x)dx = 1. \tag{3}$$

Let,

$$\phi_k^j(x) = 2^{-\frac{j}{2}}\phi(2^{-j}x-k), \tag{4}$$

and

$$\psi_k^j(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x-k), \tag{5}$$

where $j, k \in Z$, denote the dilations and translations of the scaling function and the wavelet.

The coefficients $H = \{h_k\}_{k=0}^{L-1}$ and $G = \{g_k\}_{k=0}^{L-1}$ are related by $g_k = (-1)^k h_{L-k}$ for $k = 0, ..., L-1$. Furthermore, H and G are chosen so that dilations and translations of the wavelet, $\psi_k^j(x)$, form an orthonormal basis of $L^2(R)$ and so that $\psi(x)$ has M vanishing moments. In other words, $\psi_k^j(x)$ will satisfy

$$\delta_{kl}\delta_{jm} = \int_{-\infty}^{\infty} \psi_k^j(x)\psi_l^m(x)dx, \tag{6}$$

where δ_{kl} is the Kronecker delta function. Also, $\psi(x) = \psi_0^0(x)$ satisfies

$$\int_{-\infty}^{\infty} \psi(x) x^m dx = 0, \tag{7}$$

for $m = 0, ..., M-1$. Under the conditions of the previous two equations, for any function $f(x) \in L^2(R)$ there exists a set $\{d_{jk}\}$ such that

$$f(x) = \sum_{j \in Z} \sum_{K \in Z} d_{jk} \psi_k^j(x), \tag{8}$$

where

$$d_{jk} = \int_{-\infty}^{\infty} f(x) \psi_k^j(x) dx. \tag{9}$$

The number of vanishing moments of the wavelet $\psi(x)$ defines the accuracy of approximation. The two sets of coefficients H and G are known in signal processing literature as quadrature mirror filters [5]. For Daubechies wavelets the number of coefficients in H and G, or the length of the filters H and G, denoted by L, is related to the number of vanishing moments M by $2M = L$. For example, the famous Haar wavelet is found by defining H as $h_0 = h_1 = 1$. For this filter, H, the solution to the dilation equation (1), $\phi(x)$, is the box function: $\phi(x) = 1$ for $x \in [0, 1]$ and $\phi(x) = 0$ otherwise. The Haar function is very useful as a learning tool, but it is not very useful as a basis function on which to expand another function for the important reason that it is not differentiable. The coefficients, H, needed to define compactly supported wavelets with a higher degree of regularity can be found in [4]. As is expected, the regularity increases with the support of the wavelet. The usual notation to denote a Daubechies wavelet defined by coefficients H of length L is D_L.

It is usual to let the spaces spanned by $\phi_k^j(x)$ and $\psi_k^j(x)$ over the parameter k, with j fixed, to be denoted by V_j and W_j respectively:

$$V_j = \underset{k \in Z}{span}\ \phi_k^j(x), \tag{10}$$

$$W_j = \underset{k \in Z}{span}\ \psi_k^j(x). \tag{11}$$

The spaces V_j and W_j are related by [4]

$$... \subset V_1 \subset V_0 \subset V_{-1} \subset ..., \tag{12}$$

and that

$$V_j = V_{j+1} \bigoplus W_{j+1}. \tag{13}$$

The previously stated condition that the wavelets form an orthonormal basis of $L^2(R)$ can now be written as,

$$L^2(R) = \bigoplus_{j \in Z} W_j. \tag{14}$$

Two final properties of the spaces V_j are that

$$\bigcap_{j \in Z} V_j = \{0\}, \tag{15}$$

and

$$\bigcup_{j \in Z} V_j = L^2(R). \tag{16}$$

Of course, infinite sums and unions are meaningless when one begins to implement a wavelet expansion on a computer. In some way one must limit the range of the scale parameter j and the location parameter k. Consider first the scale parameter j. As stated above, the wavelet expansion is complete: $L^2(R) = \bigoplus_{j \in Z} W_j$. Therefore, any $f(x) \in L^2(R)$ can be written as,

$$f(x) = \sum_{j \in Z} \sum_{k \in Z} d_k^j \psi_k^j(x),$$

where due to orthonormality of the wavelets $d_k^j = \int_{-\infty}^{\infty} f(x) \psi_k^j(x)$. In this expansion, functions with arbitrarily small-scale structures can be represented. In practice, however, there is a limit to how small the smallest structure can be. This would depend, for example, on how fine the grid is in a numerical computation scenario or perhaps what the sampling frequency is in a signal processing scenario. Therefore, on a computer an expansion would take place in a space such as

$$V_0 = W_1 \oplus W_2 \oplus \ldots \oplus W_J \oplus V_J, \tag{17}$$

and would appear as,

$$P_{V_0} f(x) = \sum_{k \in Z} s_k^J \phi_k^J(x) + \sum_{j=1}^{J} \sum_{k \in Z} d_k^j \psi_k^j(x), \tag{18}$$

where again due to orthonormality of the basis functions $d_k^j = \int_{-\infty}^{\infty} f(x)\psi_k^j(x)$, and $s_k^J = \int_{-\infty}^{\infty} f(x)\phi_k^J(x)$. In this expansion, scale $j = 0$ is arbitrarily chosen as the finest scale that is needed, and scale J would be the scale at which a kind of local average, $\phi_k^J(x)$, provides sufficient large scale information. In language that is likely to appeal to the electrical engineer it can be said that $\phi_k^J(x)$ represents the direct current portion of a signal at location k and that $\psi_k^j(x)$ represents the alternating current portion of a signal at, very roughly, frequency j and location k. As stated above, one must also limit the range of the location parameter k If one assumes periodicity, then the periodicity of $f(x)$ induces periodicity on all wavelet coefficients, s_k^j and d_k^j, with respect to k. Without periodicity, the location parameter k will begin at 1 with the left-hand side boundary functions and end with some maximum number N at the right-hand side boundary functions.

This completes the definition of wavelets.

3 Finite Difference Grid Refinement and Wavelets

In this section it will be shown that Daubechies-based wavelet methods when considered in the physical space are equivalent to explicit finite difference methods with grid refinement. In a Daubechies wavelet method the 'refinement' is accomplished by adding wavelet bases functions in regions where structure exists corresponding to the scale of the wavelet used for analysis. In a finite difference method the 'refinement' is accomplished by adding grid points in regions chosen by some grid refinement mechanism. In this section it is argued that since wavelet methods correspond to central finite difference operators when the grid is uniform and since wavelet methods contain a natural and effortless mechanism for 'grid refinement', then one can simply use the wavelets to refine a grid for finite difference operators. In this way one can maintain the superconvergence encountered with periodic boundary conditions, see [7], which is lost when one constructs wavelets on an interval, see [9]. That is, boundary conditions are imposed in the same manner as for finite difference operators. Furthermore, there is no longer a difficulty with nonlinear terms requiring constant transformation between the physical space and the coefficient space since all calculations are done in the physical space.

This section contains four subsections:

1. The wavelet decomposition matrix will be constructed.

2. It will be seen that under the assumption of periodicity and without data compression that the effect in the physical space of differentiation in the D_4 subspace V_0 is exactly the same as differentiation with the optimal central 4th-order finite difference operator.

3. Now we compare wavelets and finite difference in the subspace $V_0 = W_1 \oplus V_1$. If Δx is the grid spacing in V_0 then $2\Delta x$ is the grid spacing in V_1 and the wavelet coefficients in W_1 indicate if refinement is needed for a local grid spacing of Δx.

4. Finally, a division of the subspace V_0 which might be used in practice is studied: $V_0 = W_1 \oplus W_2 \oplus W_3 \oplus V_3$. Similar to above, the grid spacing in the subspace V_3 would be $8\Delta x$. The first refinement is controlled by the subspace W_3 which can refine locally to a grid spacing $4\Delta x$. Likewise, the subspace W_2 refines locally to a grid spacing of $2\Delta x$ and W_1 to a local grid spacing of Δx.

3.1 Wavelet Decomposition Matrix

The wavelet decomposition matrix is the matrix embodiment of the dilation equation defining the scaling function and the accompanying equation defining the wavelet. The following two recursion relations for the coefficients s_k^j and d_k^j can be found from equations (1) and (2), respectively:

$$s_k^j = \sum_{n=1}^{n=2M} h_n s_{n+2k-2}^{j-1}, \tag{19}$$

and

$$d_k^j = \sum_{n=1}^{n=2M} g_n s_{n+2k-2}^{j-1}. \tag{20}$$

Denote the decomposition matrix embodied by these two equations, assuming periodicity, by $P_{N\times N}^{j,j+1}$ where the matrix subscripts denote the size of the matrix, and the superscripts indicate that P is decomposing from scaling function coefficients at scale j to scaling function and wavelet function coefficients at scale $j+1$. Let $\vec{s}_j$ contain the scaling function coefficients at scale j. (Note that when vector notation is used the scale is given as a subscript.) P therefore maps $\vec{s}_j$ onto $\vec{s}_{j+1}$ and $\vec{d}_{j+1}$:

$$P_{N\times N}^{j,j+1} : \left[\begin{array}{c} \vec{s}_j \end{array} \right] \rightarrow \left[\begin{array}{c} \vec{s}_{j+1} \\ \vec{d}_{j+1} \end{array} \right]. \tag{21}$$

Note that the vectors at scale $j+1$ are half as long as the vectors as scale j. To illustrate further, suppose the wavelet being used is the four coefficient D_4 wavelet, and suppose one wants to project from 8 scaling function coefficients at scale j to 4 scaling function coefficients at scale $j+1$ and 4 wavelet coefficients at scale $j+1$.

The decomposition matrix in this case is,

$$P_{8\times 8}^{j,j+1} \equiv \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_1 & h_2 & h_3 & h_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & h_4 \\ h_3 & h_4 & 0 & 0 & 0 & 0 & h_1 & h_2 \\ g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_1 & g_2 & g_3 & g_4 \\ g_3 & g_4 & 0 & 0 & 0 & 0 & g_1 & g_2 \end{bmatrix}, \tag{22}$$

where the periodicity is seen from the coefficients 'wrapping around'.

Now let us consider differentiation. Let the four matrices $A_{N\times N}^j$, $B_{N\times N}^j$, $C_{N\times N}^j$, and $R_{N\times N}^j$, see [7] and [1], contain the derivative projection coefficients,

$$A^j : \vec{d}_j \to \vec{\acute{d}}_j,$$

$$B^j : \vec{s}_j \to \vec{\acute{d}}_j,$$

$$C^j : \vec{d}_j \to \vec{\acute{s}}_j,$$

$$R^j : \vec{s}_j \to \vec{\acute{s}}_j,$$

where $\vec{\acute{s}}_j$ and $\vec{\acute{d}}_j$ denote the coefficients of the expansion of the derivative of a function which is initially defined by the expansion coefficients $\vec{s}_j$ and $\vec{d}_j$. The exact form of the matrices A, B, and C is not important for the discussion here. The important point is the form of the matrix R. It is always a finite difference operator. For the D_4 wavelet R corresponds to the optimal central 4th-order finite difference operator. For higher order wavelets, D_6, D_8, etc., R is a finite difference operator, but it is not optimal in the sense of using the minimum number of coefficients to obtain a given accuracy. The numerical values of the coefficients were found in [1] and the superconvergence accuracy was proven in general in [7]. For the D_4 wavelet an explicit 8×8 example

of matrix R is,

$$R_{8\times 8} = \begin{bmatrix} 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} \\ -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 & \frac{1}{12} \\ \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 \\ 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 \\ 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 \\ 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} \\ -\frac{1}{12} & 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{12} & 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 \end{bmatrix}. \tag{23}$$

We will now see how 'grid refinement' is accomplished in a wavelet scenario by examining three divisions of the subspace V_0 in three following three subsections.

3.2 Wavelet Expansion and Derivative in V_0

One can calculate the derivative of a wavelet expansion at any level in the wavelet decomposition. This subsection will explore the first of three of the options. To be explicit, suppose that a periodic function $f(x)$ has been approximated on a grid with 16 scaling function coefficients to get $\vec{s}_0$, and for the current argument assume that the coefficients have been calculated exactly. Note that periodicity of $f(x)$ induces periodicity on the coefficients $\vec{s}_0$. The coefficients of the expansion of $\frac{d}{dx}f(x)$ in V_0 are found from $\vec{s}_0$ by an application of the matrix $R^0_{16\times 16}$:

$$\begin{bmatrix} s^0_1 \\ s^0_2 \\ s^0_3 \\ s^0_4 \\ s^0_5 \\ s^0_6 \\ s^0_7 \\ s^0_8 \\ s^0_9 \\ s^0_{10} \\ s^0_{11} \\ s^0_{12} \\ s^0_{13} \\ s^0_{14} \\ s^0_{15} \\ s^0_{16} \end{bmatrix} \xrightarrow{\frac{1}{\Delta x}R^0_{16\times 16}} \begin{bmatrix} s'^0_1 \\ s'^0_2 \\ s'^0_3 \\ s'^0_4 \\ s'^0_5 \\ s'^0_6 \\ s'^0_7 \\ s'^0_8 \\ s'^0_9 \\ s'^0_{10} \\ s'^0_{11} \\ s'^0_{12} \\ s'^0_{13} \\ s'^0_{14} \\ s'^0_{15} \\ s'^0_{16} \end{bmatrix}. \tag{24}$$

Let us now examine the entire process of going from point values in the physical space to scaling function coefficients in V_0, differentiating, and finally returning to point values of the differentiated function in the physical space. Begin with $f(x) \in L^2(R)$ defined at 16 evenly-spaced points over $[-\pi, \pi)$ and let $f(x)$ is 2π periodic. To differentiate the samples of $f(x)$, $\vec{f}$, with the 4-th order optimal central finite difference operator, say D_{fd4}, we get,

$$\vec{f}' = D_{fd4}\vec{f}. \tag{25}$$

Now, suppose that we have mapped these 16 samples into the scaling function coefficients in V_0 by applying the circular, periodicity implies circularity, see [7], quadrature matrix Q,

$$\vec{s}_0 = Q_{16\times 16}\vec{f}. \tag{26}$$

We now find $\vec{s}_0'$ by applying $\frac{1}{\Delta x}R_{16\times 16}$. The two matrices R and Q are, however, both circular and, hence, commute:

$$\vec{s}_0' = Q_{16\times 16}\frac{1}{\Delta x}R_{16\times 16}\vec{f}. \tag{27}$$

Now, returning to the physical space we get,

$$\vec{f}' = Q^{-1}_{16\times 16}Q_{16\times 16}\frac{1}{\Delta x}R_{16\times 16}\vec{f}, \tag{28}$$

and we are back to equation (25) again since,

$$D_{fd4} \equiv \frac{1}{\Delta x}R_{16\times 16}. \tag{29}$$

Hence, we have shown that under the assumption of periodicity and without data compression that the D_4 wavelet differentiation corresponds exactly to optimal central 4th-order finite differencing. Note that data compression is the goal of any wavelet method. The embodiment of data compression in the physical space is a nonuniform grid. That is, the grid must be dense in regions where small structure requires fine resolution and the grid can be sparse when the data is composed of large scale components.

Now we move to the first decomposition of $V_0 = W_1 \oplus V_1$ in which data compression can be achieved.

3.3 Wavelet Expansion and Derivative in $W_1 \oplus V_1$

Consider now a decomposition of the vector of scaling function coefficients $\vec{s}_0$ onto the scaling function and wavelet coefficients at scale $j = 1$ by an application of the matrix $P^{0,1}_{16\times 16}$:

$$\begin{bmatrix} s^0_1 \\ s^0_2 \\ s^0_3 \\ s^0_4 \\ s^0_5 \\ s^0_6 \\ s^0_7 \\ s^0_8 \\ s^0_9 \\ s^0_{10} \\ s^0_{11} \\ s^0_{12} \\ s^0_{13} \\ s^0_{14} \\ s^0_{15} \\ s^0_{16} \end{bmatrix} \xrightarrow{P^{0,1}_{16\times 16}} \begin{bmatrix} s^1_1 \\ s^1_2 \\ s^1_3 \\ s^1_4 \\ s^1_5 \\ s^1_6 \\ s^1_7 \\ s^1_8 \\ d^1_1 \\ d^1_2 \\ d^1_3 \\ d^1_4 \\ d^1_5 \\ d^1_6 \\ d^1_7 \\ d^1_8 \end{bmatrix} . \tag{30}$$

As in V_0, we have 16 basis functions, but now the subspace V_0 is decomposed into 'low frequency', V_1, and 'high frequency', W_1, components: $V_0 = V_1 \oplus W_1$. In order to calculate the coefficients of the derivative expansion in $V_1 \oplus W_1$ the following projections are calculated:

$$\begin{bmatrix} \vec{s}_1 \\ \vec{d}_1 \end{bmatrix} = \frac{1}{2\Delta x} \begin{bmatrix} R^1_{8\times 8} & C^1_{8\times 8} \\ B^1_{8\times 8} & A^1_{8\times 8} \end{bmatrix} \cdot \begin{bmatrix} \vec{s}_1 \\ \vec{d}_1 \end{bmatrix} . \tag{31}$$

If one now applies the matrix $(P^{0,1}_{16\times 16})^T$ (T denotes transpose and hence inverse for this unitary matrix) to the derivative coefficients at scale $j = 1$ one gets,

$$\begin{bmatrix} \vec{s}_0 \end{bmatrix} = (P^{0,1}_{16\times 16})^T \cdot \begin{bmatrix} \vec{s}_1 \\ \vec{d}_1 \end{bmatrix} , \tag{32}$$

and one gets exactly the same coefficients as before when the matrix $\frac{1}{\Delta x} R^0_{16\times 16}$ was applied to $\vec{s}_0$.

Now suppose that $f(x)$ is smooth enough such that a grid of eight points provides sufficient resolution. Define $\vec{f}_2$ to be the 8 element vector containing every other entry of the 16 element vector $\vec{f}$. 4-th order differentiation of $\vec{f}_2$ is performed by applying $\frac{1}{2\Delta x} R_{8\times 8}$,

$$\vec{\dot{f}}_2 = \frac{1}{2\Delta x} R_{8\times 8} \vec{f}_2. \tag{33}$$

Similar to above, we project the eight dimensional $\vec{f}_2$ into the eight dimensional wavelet subspace V_1 using $Q_{8\times 8}$ and differentiate to get,

$$\vec{s}_1 = \frac{1}{2\Delta x} R_{8\times 8} Q_{8\times 8} \vec{f}_2, \tag{34}$$

followed by projection back into the physical space with the matrix $Q^{-1}_{8\times 8}$ we get equation (33) again.

That is, we have seen that if we work only in V_0 that we have 4th-order finite differencing with a grid spacing of Δx, whereas if we work only in V_1 we have 4th-order finite differencing with a grid spacing of $2\Delta x$. But, the two subspaces V_0 and V_1 are related by $V_0 = V_1 \oplus W_1$. Recall, that the subspace W_1 contains bases functions which are locally oscillatory and are compactly supported. An inner product of these bases with the data $f(x)$ will detect local oscillations in $f(x)$ and provide exactly the information necessary to refine the grid locally from $2\Delta x$ to Δx. This wavelet grid refinement mechanism can be used not only to add wavelet bases functions where one has a large inner product but also to refine the grid in the same region and at a scale corresponding to the wavelet scale.

3.4 Wavelet Expansion and Derivative in $W_1 \oplus W_2 \oplus W_3 \oplus V_3$

Let us close this section with a wavelet decomposition that one might use in practice. That is, again V_0 denotes the finest scale subspace and we decompose V_0 as,

$$V_0 = W_1 \oplus W_2 \oplus W_3 \oplus V_3. \tag{35}$$

The vector of coefficients in this subspace is obtained by the following decompositions:

$$\begin{bmatrix} s_1^0 \\ s_2^0 \\ s_3^0 \\ s_4^0 \\ s_5^0 \\ s_6^0 \\ s_7^0 \\ s_8^0 \\ s_9^0 \\ s_{10}^0 \\ s_{11}^0 \\ s_{12}^0 \\ s_{13}^0 \\ s_{14}^0 \\ s_{15}^0 \\ s_{16}^0 \end{bmatrix} \xrightarrow{P^{0,1}_{16\times 16}} \begin{bmatrix} s_1^1 \\ s_2^1 \\ s_3^1 \\ s_4^1 \\ s_5^1 \\ s_6^1 \\ s_7^1 \\ s_8^1 \\ d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \\ d_5^1 \\ d_6^1 \\ d_7^1 \\ d_8^1 \end{bmatrix} \xrightarrow{P^{1,2}_{8\times 8}} \begin{bmatrix} s_1^2 \\ s_2^2 \\ s_3^2 \\ s_4^2 \\ d_1^2 \\ d_2^2 \\ d_3^2 \\ d_4^2 \\ d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \\ d_5^1 \\ d_6^1 \\ d_7^1 \\ d_8^1 \end{bmatrix} \xrightarrow{P^{2,3}_{4\times 4}} \begin{bmatrix} s_1^3 \\ s_2^3 \\ d_1^3 \\ d_2^3 \\ d_1^2 \\ d_2^2 \\ d_3^2 \\ d_4^2 \\ d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \\ d_5^1 \\ d_6^1 \\ d_7^1 \\ d_8^1 \end{bmatrix} . \tag{36}$$

Let us suppose that we have performed the above wavelet decomposition on a vector of data points, $\vec{f}$, at some point during a simulation which contains data at many different scales. Furthermore, let there be a shock, or a near shock, near the right-hand boundary. The coefficients s_1^3 and s_2^3 represent local averages in the subspace V_3 corresponding to the 'base grid' of size $8\Delta x$ and will not yield much useful information with respect to the shock. The coefficients d_1^3 and d_2^3 of the subspace W_3 will yield the presence of oscillations of relatively large scale. A true shock contains all frequencies and one would expect to have some coefficient perturbation even in W_3 yielding grid refinement to a grid spacing of $4\Delta x$ in a neighborhood of the shock. The coefficients d_1^2, d_2^2, d_3^2, and d_4^2 in the subspace W_2 will detect oscillations at the corresponding scale only near the shock. That is, the first two coefficients d_1^2 and d_2^2 are responsible for detecting small scale structure at the left-hand side of the domain which is away from the shock, and we, therefore, expect that they will be near zero in magnitude. The coefficients d_3^2 and d_4^2, on the other hand, are positioned near the shock and will have a relatively large amplitude indicating the presence of small scales. The grid will, therefore, be refined to a spacing of $2\Delta x$ at the right-hand side of the domain. Likewise, the remaining coefficients in the subspace W_1 will refine the

grid to a spacing of Δx at the right-hand side of the domain.

In conclusion, this section has been devoted to first illustrating how Daubechies-based wavelet methods are in essence finite difference methods with grid refinement, and second to illustrating how the Daubechies-based wavelets can be used to define a grid for finite difference methods. The next section will make this symbiotic relationship between Daubechies wavelets and finite difference methods formal.

4 The Wavelet-Optimized, Adaptive Grid Finite Difference Method, (WOFD)

The new method is to apply finite difference on a grid which is defined by the magnitude of wavelet coefficients at various scales. That is, wavelets can detect oscillations in a function at any location and scale. Given a function $f(x)$ for $x \in I$, where I is some interval, one decomposes $f(x)$ into a set of wavelet coefficients which depend on two parameters, one for location and one for scale, say d_k^j, where k is the location parameter and j is the scale parameter. If a wavelet coefficient is large in magnitude,

$$|d_k^j| > T, \tag{37}$$

or large in energy (In practice the two criteria yield roughly the same grid.),

$$(d_k^j)^2 > T, \tag{38}$$

where T is a coefficient threshold chosen by the user, then WOFD adds a grid point, or two, at location k and at a grid density corresponding to the scale j. That is, WOFD defines a grid which will completely resolve a function across the entire domain without over resolving it where it is relatively smooth, or composed only of large scale structure. For the specific case of the D_4 wavelet outlined in the previous section, the D_4 wavelet decomposition provides the optimal grid for 4th-order finite differencing.

The grid definition should be made by a Daubechies wavelet which corresponds in terms of superconvergence accuracy to the accuracy of the finite difference operator. That is, it was proven in [7] that the differentiation matrix for the Daubechies wavelet D_{2M}, where M is the number of vanishing moments, displays differentiation accuracy of order $2M$ under the assumptions of periodicity and a uniform grid. Recall, that this wavelet subspace can only represent exactly the first M polynomials as determined by the number of vanishing moments. This order of accuracy $2M$ should equal the order of accuracy of the finite difference operator for optimal grid selection.

In the next section WOFD will be applied to Burgers' equation.

5 WOFD Applied to Burgers' Equation

In this section WOFD will be applied to Burgers' equation,

$$\frac{\partial U}{\partial t} = -U\frac{\partial U}{\partial x} + \epsilon\frac{\partial^2 U}{\partial x^2}, \tag{39}$$

with the initial condition,

$$u(x,0) = \frac{1}{3} + \frac{2}{3}\sin(2\pi x). \tag{40}$$

The goal of this section is to illustrate that WOFD using the D_4 wavelet produces a solution on a nonuniform reduced grid which is 'equivalent in character' to the solution provided by 4th order finite differencing on the finest uniform grid. That is, for a given viscosity, ϵ, there exists a grid size fine enough such that oscillations do not develop at the 'shock'. This can be made more precise by saying that one has a grid fine enough such that the total variation of the solution does not increase. 'Equivalent in character' means that the total variation of the solution provided by WOFD increases if and only if the total variation of the solution produced by finite differencing on the finest uniform grid increases.

In all the following plots the uniform finite differencing is provided by the optimal central 4th-order finite difference operator. The temporal discretization is achieved by 4th-order Runge-Kutta. The WOFD coefficient threshold which determines which grid points to use based on the wavelet coefficient magnitude is set to $T = .001$. Note that when the WOFD coefficient threshold is set to $T = 0$ that one gets finite differencing on the uniform finest grid. In addition, if the coefficient threshold is set to a very large number, say $T = 100$, then one gets finite differencing on a uniform sparse grid. The size of this sparse grid is determined by the number of wavelet decompositions one specifies.

5.1 Periodic Boundary Conditions

In figure (1) WOFD is compared to finite differencing on uniform grid sizes 32, 64, and 128. The upper left-hand plot has the WOFD solution superimposed on the finite

difference solution. The two solutions are visually indistinguishable. Along the x-axis of the plot an 'x' is placed at every position where a WOFD grid point is used. One can see that the grid points are dense at the shock and sparse where the solution is smooth.

The remaining three plots show finite difference solutions on various uniform grid sizes. One sees oscillations for grid sizes 32 and 64 but not for grid size 128.

Figure (4) provides an additional plot for periodic boundary conditions with a slightly larger viscosity.

5.2 Nonperiodic Boundary Conditions

The boundary conditions considered here are such that the boundary values of the solution are required to be fixed at the initial condition values,

$$u(0,t) = u(1,t) = \frac{1}{3}. \tag{41}$$

In all the plots, differentiation at the boundary for the uniform finite difference method is achieved by the optimal 4th order one sided finite difference coefficients, see [2]. For WOFD both 4th order and 3rd order boundary differentiation will be considered.

In Figure (2) the differentiation at the boundary is 4th order, and, as above, WOFD provides a solution which is 'equivalent in character' to the finite difference solution on the finest grid, 128 grid points, while reducing the number of degrees-of-freedom necessary to achieve this solution.

In Figure (3) the differentiation at the boundary is 3rd order. The solution for the 3rd-order boundary differentiation is good, but a slight difference can be seen with the finite difference method on the finest grid. Again, finite difference on more coarse grids oscillates more at the shock than the finest grid solution.

Figure (5) provides an additional plot illustrating the solution provided by WOFD for the nonperiodic case.

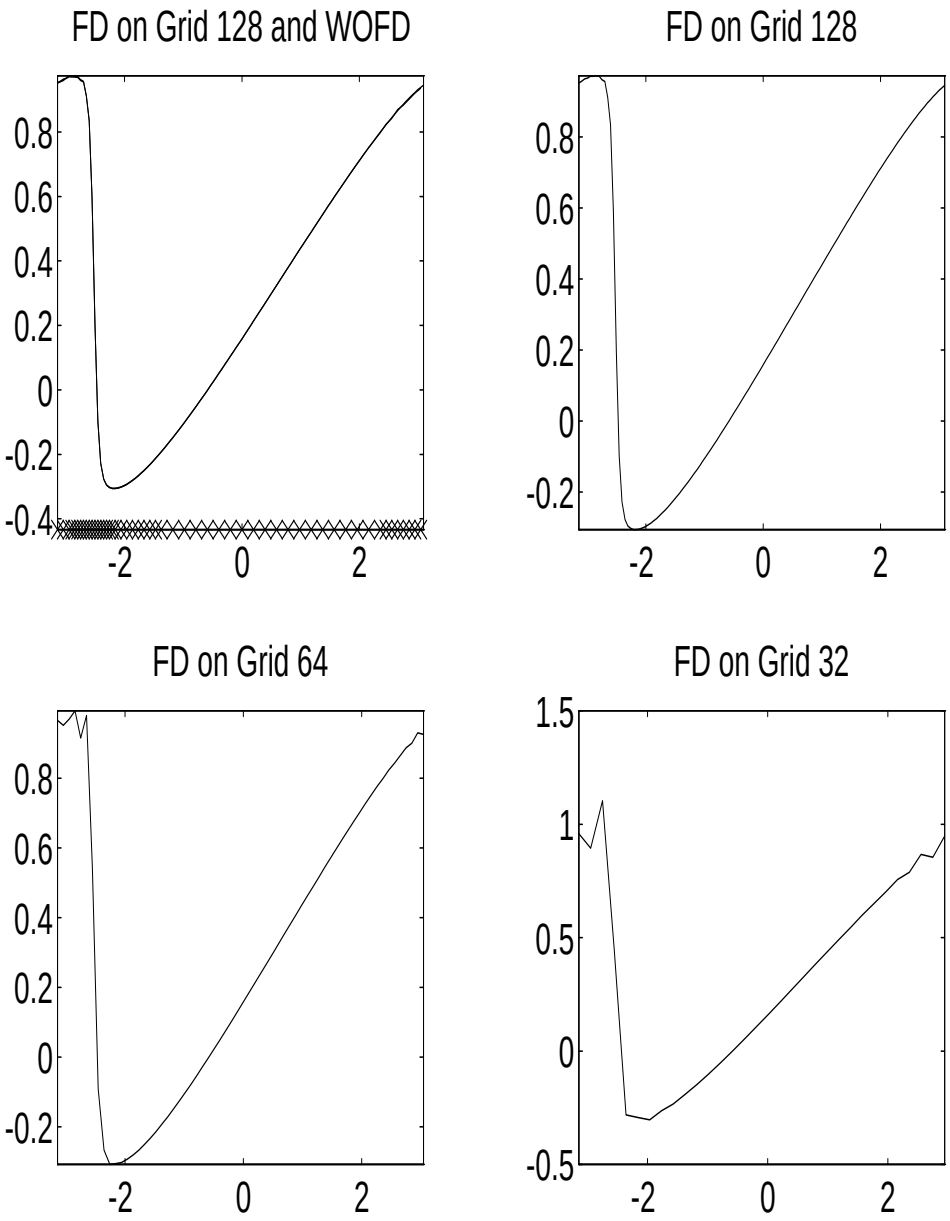

Figure 1: Illustration of equivalence of WOFD to an equivalent order finite difference method applied across the entire domain at the finest scale. The boundary conditions are periodic. Final time = 2, Viscosity = .02.

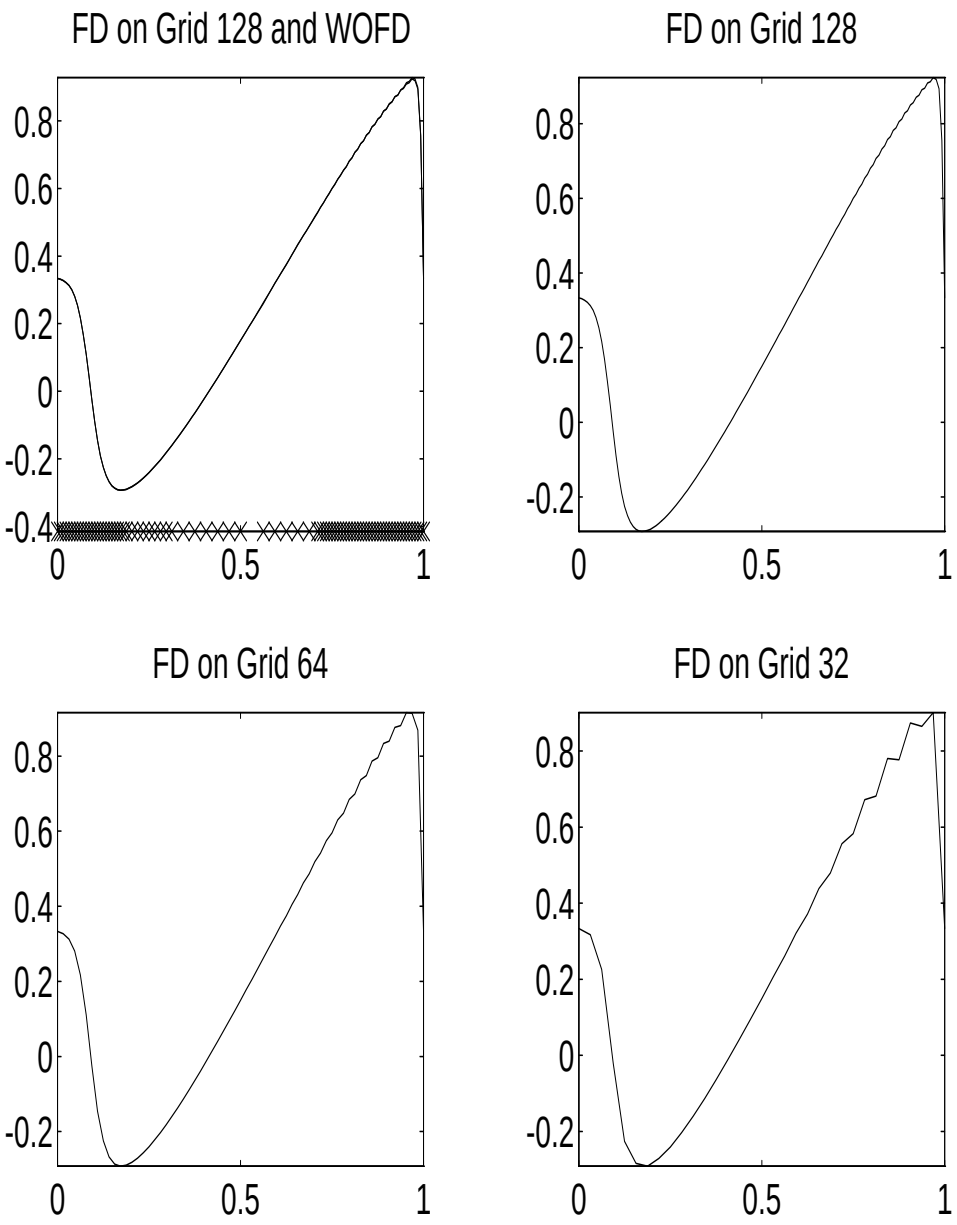

Figure 2: Illustration of equivalence of WOFD to an equivalent order finite difference method applied across the entire domain at the finest scale. The boundary values are fixed at the initial condition values. Differentiation at the boundary is 4th order. Final time = .3, Viscosity = .005.

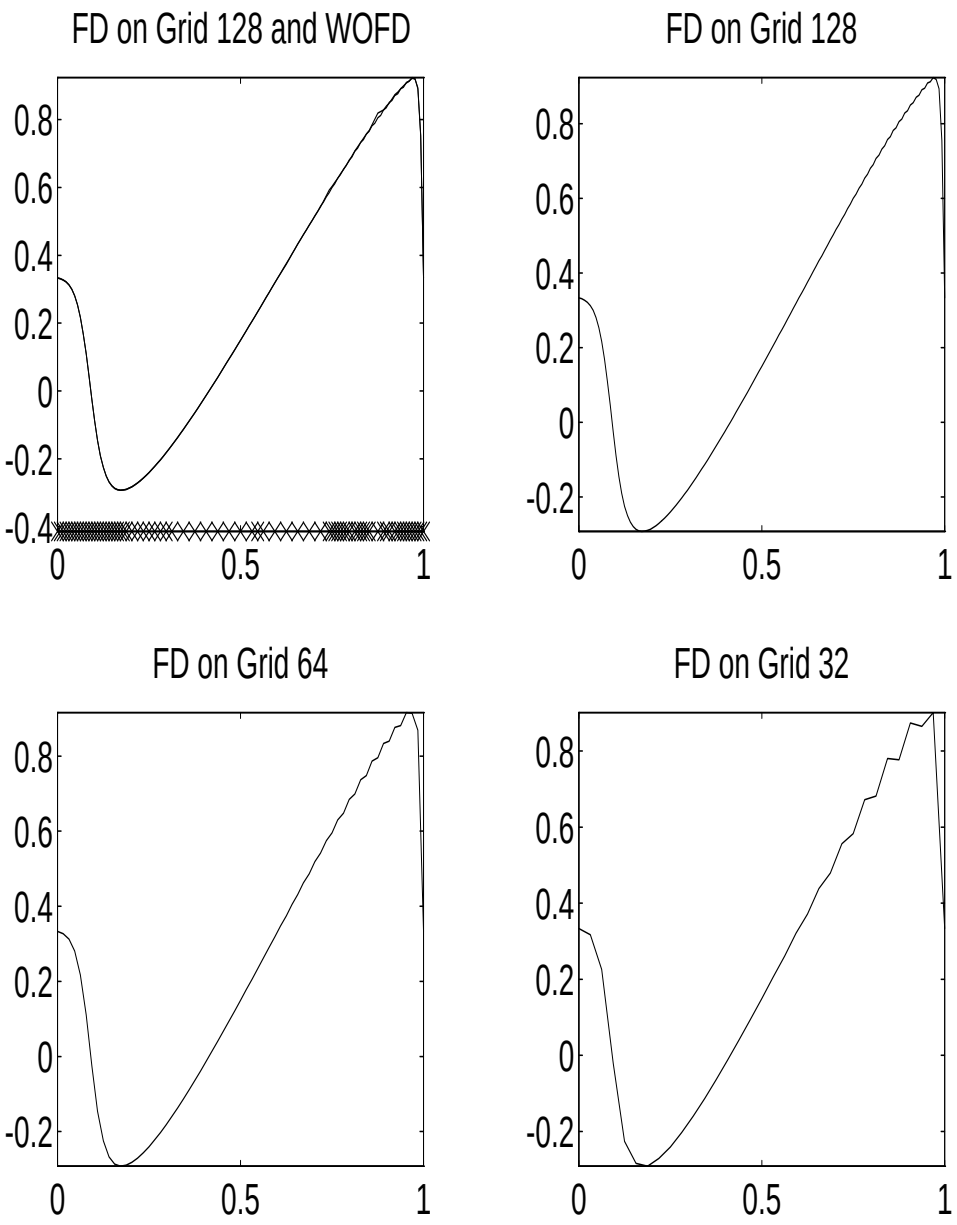

Figure 3: Illustration of equivalence of WOFD to an equivalent order finite difference method applied across the entire domain at the finest scale. The boundary values are fixed at the initial condition values. Differentiation at the boundary is 3rd order. Final time = .3, Viscosity = .005.

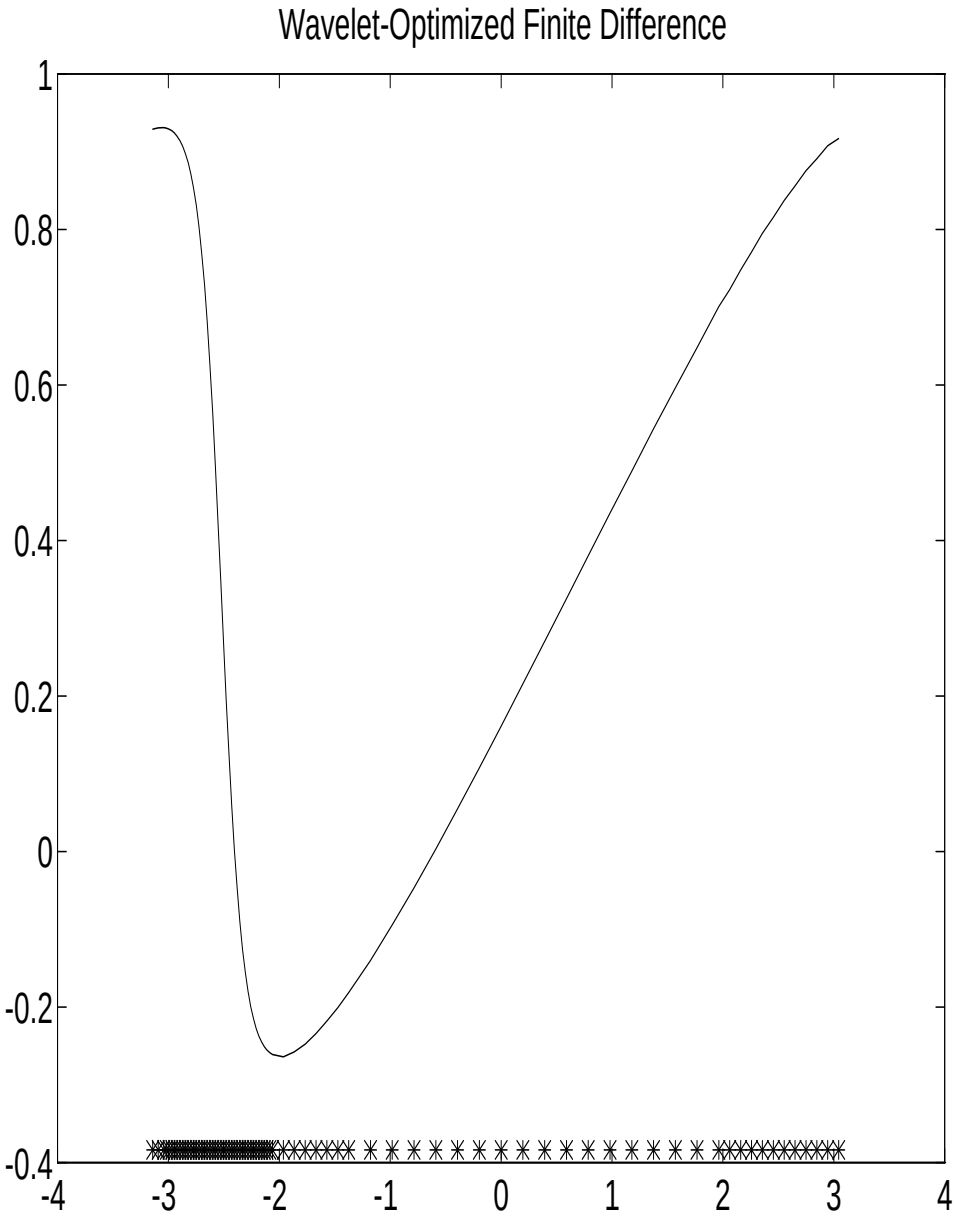

Figure 4: WOFD applied to Burgers equation. Boundary conditions are periodic, final time is 2, viscosity is .05.

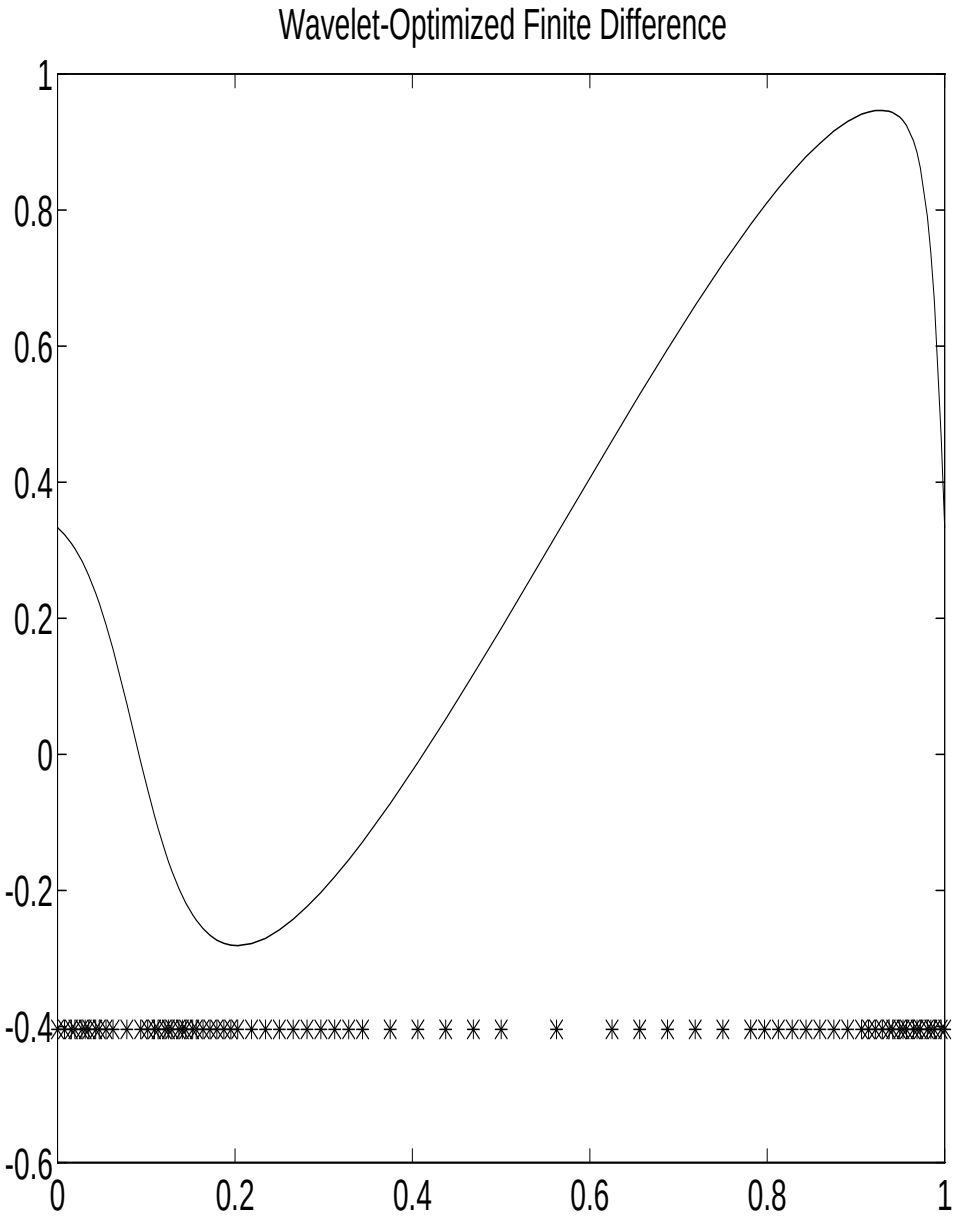

Figure 5: WOFD applied to Burgers equation. Boundary values are fixed to initial condition values, final time is .3, viscosity is .02.

6 Accuracy of WOFD

In this section the order of accuracy will be examined. For numerical methods where the grid is uniform the order of accuracy is clearly defined. For WOFD, on the other hand, the discussion of accuracy is slightly more complicated. That is, it can be said that WOFD approximates a 4th-order finite difference method as well as one desires, and when the coefficient threshold is set to zero then WOFD is truly 4th-order. So, WOFD approximates methods of a given order as well as is desired. In addition, it will be seen that the WOFD method has a very nice feature that the rate of growth of the error in approximating the derivative is at most a linear function of frequency.

6.1 Error in Derivative Approximation

The finite difference equations used in this paper use either a 4-point stencil or a 5-point stencil. The derivative approximation error for the equations on a 3-point stencil will be given. The derivative approximation error for a larger stencil is an obvious extension of the error given here.

Consider the following Lagrangian interpolation of a quadratic polynomial through the three points: $(x_1, f(x_1))$, $(x_2, f(x_2))$, and $(x_3, f(x_3))$, for $x_1 < x_2 < x_3$:

$$g(x) = \tag{42}$$
$$f(x_1)\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + f(x_2)\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + f(x_3)\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}.$$

If we differentiate $g(x)$ and evaluate at x_2 we get,

$$\frac{d}{dx}g(x)|_{x_2} = \frac{d}{dx}f(x)|_{x_2} + \Delta_1\Delta_2\frac{1}{6}f^{(''')}(a), \tag{43}$$

for some $a \in [x_1, x_3]$, where $\Delta_1 = x_2 - x_1$ and $\Delta_2 = x_3 - x_2$.

6.2 Control of Error Growth

As given above we will examine the special case of a 3-point stencil where the grid is refined by the Haar wavelet. In practice I never use the Haar wavelet, but it is very useful as an illustration tool.

As given above the error is,

$$Err = \Delta_1 \Delta_2 \frac{1}{6} f^{(''')}(a). \tag{44}$$

for some $a \in [x_1, x_3]$. Now, let f be a pure sinusoid of frequency ξ: $f(x) = e^{i\xi x}$. Then the magnitude of the error becomes,

$$6|Err| = \Delta_1 \Delta_2 \xi^3. \tag{45}$$

The grid refinement mechanism used by WOFD is such that, roughly,

$$\Delta = \frac{1}{\xi}. \tag{46}$$

The magnitude of the error becomes,

$$6|Err| = \xi. \tag{47}$$

That is, the refinement mechanism keeps the rate of growth of the error linear with respect to frequency. Whereas, without the grid refinement the error grows as a cubic in this case.

This is one particular refinement mechanism, but is representative of a typical refinement method.

6.3 Relationship of Threshold Size to Solution

The grid for WOFD is chosen by the size of wavelet coefficients found from a wavelet decomposition of the numerical solution at a given time. One chooses a threshold with which to measure the coefficient size. That is, if the threshold is set to .001 then the grid is refined at a given location and scale whenever the wavelet coefficients at that location and scale are larger in magnitude than .001. If the threshold is set to 0 then one gets finite difference on an evenly-spaced grid at the finest scale. The question then becomes, what is the relationship between this threshold value and the solution achieved by WOFD. As of now, a theoretical relationship does not exist but a numerical relationship does. That is, if the threshold is set to $T = 1e^{-p}$ then one

can expect that both the l_1 and l_∞ difference between WOFD and finite difference at the finest scale will be a constant times this threshold, say kT where $k < 10$. For example, for a simulation with periodic boundary conditions, viscosity set at .01, and the final time set to 2, an l_1 difference of $3.27e^{-4}$ and an l_∞ difference of $2.31e^{-3}$ were found for a threshold value of $T = 1e^{-3}$. This relationship was typical of all simulations which were run.

7 Stability of WOFD

The discussion of the stability of WOFD will be given in terms of the eigenvalues of the differentiation matrix.

7.1 Eigenvalues of Differentiation Matrix

The differentiation matrix produced by this scheme will have a full set of eigenvectors. We can, therefore, look at instability through the magnitude of the eigenvalues.

Recall that the WOFD method can produce essentially a completely arbitrary grid. The differentiation matrix can, therefore, take on an unlimited number of forms. For this reason, an analytical approach is not within reach. Therefore, experimental results which give the magnitude of the eigenvalues of the differentiation after each grid update will be given. On the following pages the eigenvalues will be given for the matrix,

$$M = I + \mathcal{D}\Delta t + 1/2\mathcal{D}^2(\Delta t)^2 + 1/6\mathcal{D}^3(\Delta t)^3 + 1/24\mathcal{D}^4(\Delta t)^4, \tag{48}$$

which corresponds to WOFD being applied to the linear equation

$$u_t = u_x,$$

with 4th-order Runge-Kutta time discretization. The grid is the grid that is chosen for the nonlinear Burgers' equation. It is seen that the magnitude of the eigenvalues do sometimes exceed 1, but they rarely exceed 1 by very much. That is, for the periodic case, considering the maximum eigenvalue for the 4th-order RK for every grid encountered, the maximum eigenvalue magnitude for the entire run up to time 2 is 1.0004. This eigenvalue is close enough to 1 in magnitude not to excite instability. In fact, the data would have to have a large component in the direction of the corresponding eigenvector and one would have to iterate 100 times to get amplification of 4% in the direction of this eigenvector.

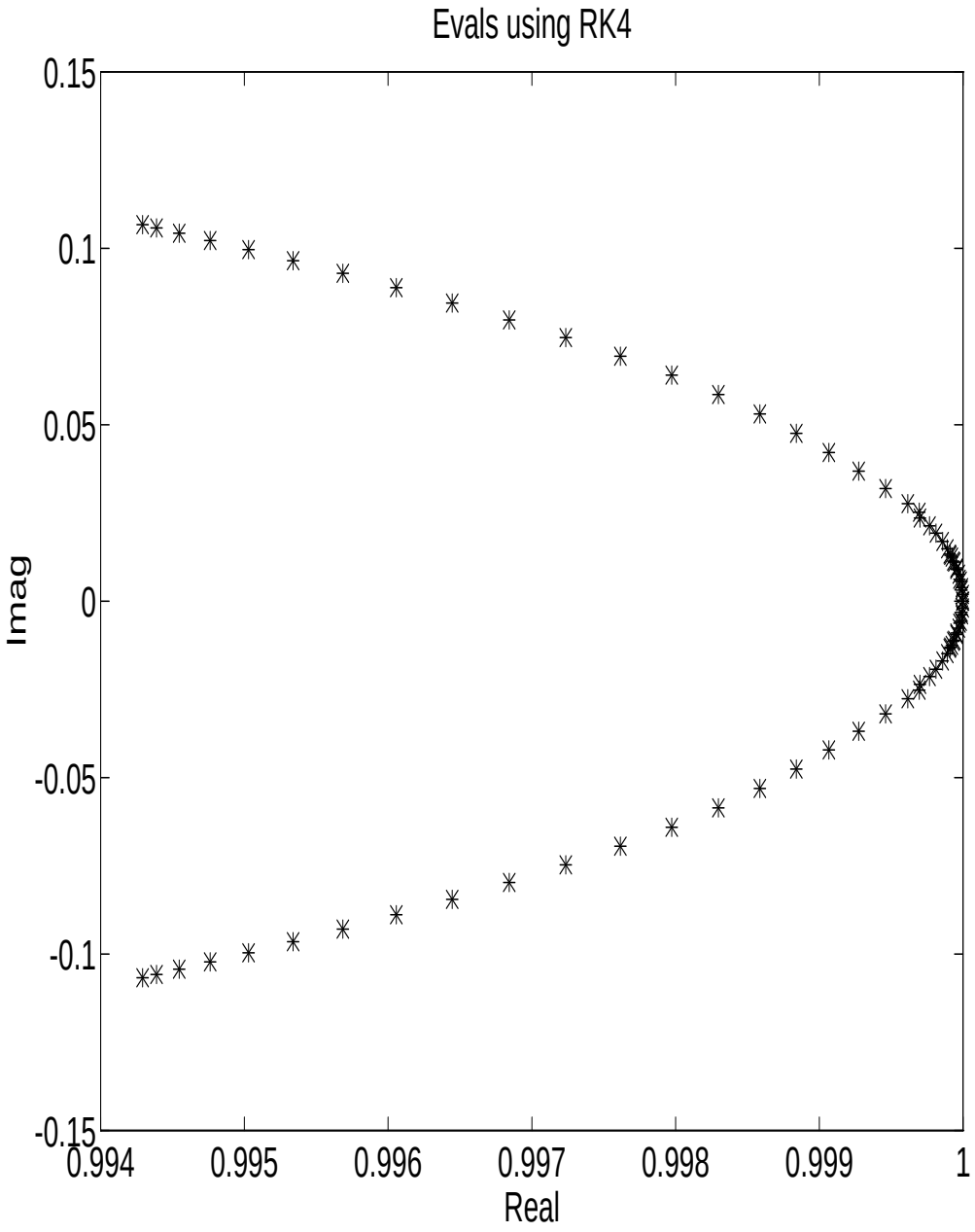

Figure 6: Eigenvalues of the 4th order Runge-Kutta differentiation matrix at time 2. The boundary conditions are periodic.

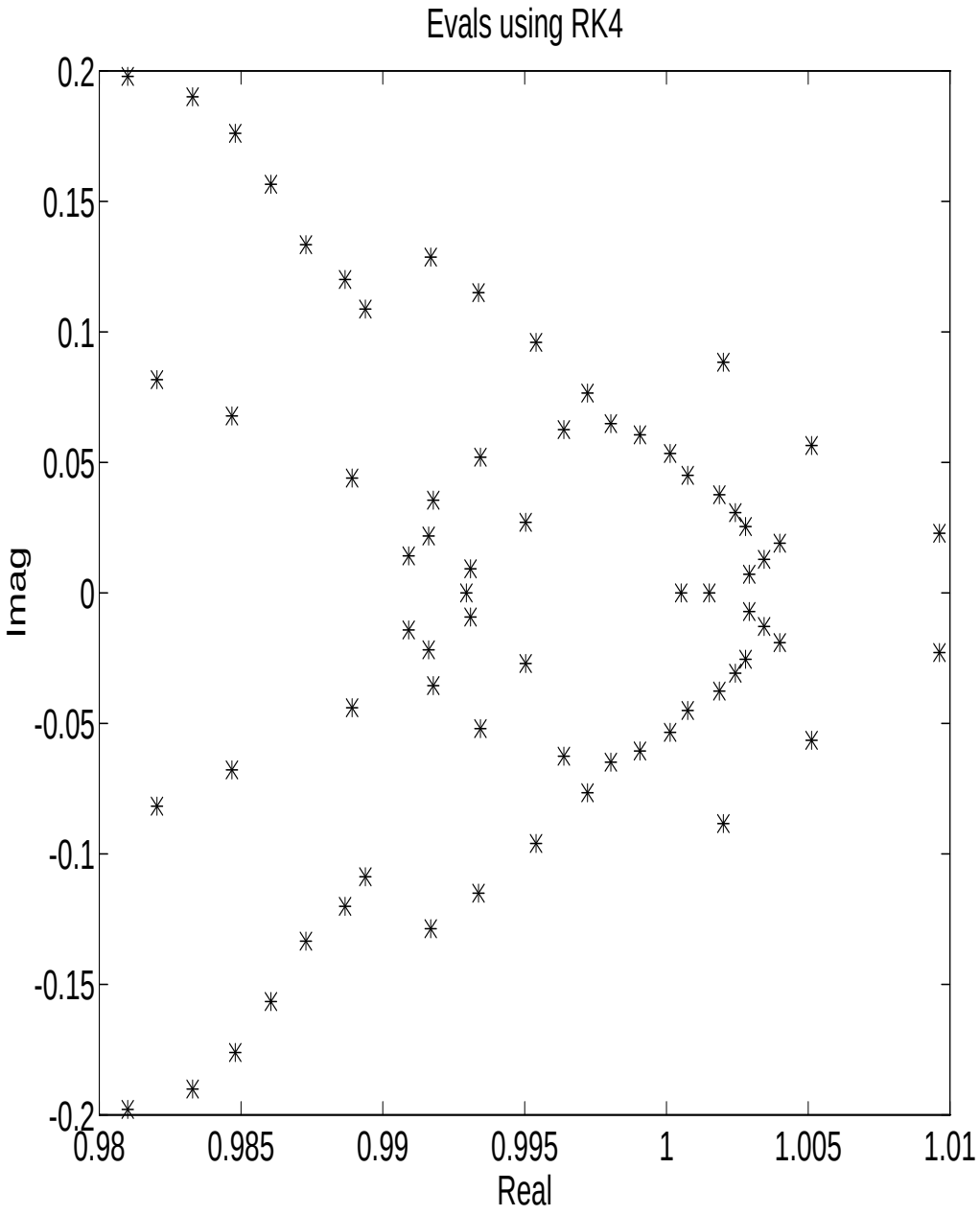

Figure 7: Eigenvalues of the 4th order Runge-Kutta differentiation matrix at time 2. The boundary values are fixed at the initial condition values.

8 Efficiency of WOFD

In a word, the efficiency of the WOFD method depends primarily on the data compression ratio. That is, we choose some finest grid which captures all the details of our solution throughout the entire run. Let's say that this finest grid has N degrees-of-freedom. Now choose how close, in $|\cdot|_2$, you desire your WOFD solution to be to the solution on the finest grid. Choosing this 'closeness parameter' dictates the data compression ratio. Let's say that the WOFD method needs only N_0 degrees-of-freedom to satisfy this criterion. Then, the amount of work done is, roughly, $\frac{N_0}{N}$ times the amount of work done to get the solution on the finest grid.

8.1 Work Involved for Grid Update

The grid update requires a number of steps. I will give a worst case scenario in estimating the number of operations.

A grid update requires order N multiplies where N is the number of degrees-of-freedom in the finest scale. The constant that is multiplied times N is reasonably large, and the following will show where the operations are used:

1. One must reconstruct the function on the finest grid. This requires about $10N$ multiplies.

2. Next, one must perform a wavelet decomposition. For a Daubechies 4 wavelet decomposition the filters are length 4. Therefore, the first decomposition requires $4N$ multiplies. Likewise, the second decomposition requires $2N$ multiplies. The number of decompositions will determine the number of multiplies, but let's say that this step, also, requires about $10N$ multiplies.

3. Choosing a grid from a wavelet decomposition does not require many operations, but it does need a number of 'IF-THEN' statements. There is roughly 1 'IF-THEN' statement for each degree-of-freedom. Let me, once again, overestimate

the cost of each 'IF-THEN' and also compensate for a few other operations that are necessary by saying this requires roughly $10N$ operations.

4. Once a grid is chosen the new differentiation matrix is found. This requires about $40N_0$ multiplies in a worst case scenario. The second derivative filter can be found from the first derivative filter by a convolution of each filter. This requires about $25N_0$ multiplies. Here, N_0 is the number grid points used in the compressed scenario. N_0 is some fraction of N.

The total number of multiplies is the sum of all the above multiples. That is, about $30N + 65N_0$. These numbers are rough, and we might as well round up to be safe and say, $50N + 100N_0$ multiplies are needed to define the grid and build a new differentiation matrix. This is reasonably expensive, but the update can be done rarely during a run. Compare this to finite difference on the finest scale. The filters for 1-st and 2-nd order differentiation are length 5. Each step of Runge-Kutta requires, therefore, at least $10N$ multiplies. If we are using a fourth order RK then we have at least $40N$ multiplies for each time step. It is, therefore, fair to say that the grid update step requires about the same amount of work as one time step taken using the full grid.

8.2 Work Saved with Larger Time Step

All the numerical scenarios use explicit time stepping. For Burgers' equation with viscosity ϵ the time step is set to,

$$\Delta t = \lambda \frac{(\Delta x)^2}{\epsilon},$$

where Δx is the minimum spacing of the grid produce by WOFD. At the beginning of any simulation if the initial condition is smooth, as measured by a wavelet decomposition, then the minimum Δx produced by WOFD is much larger than the Δx used on the finest grid. This allows a much larger time step without introducing large errors.

The larger time step means that far fewer total time steps will need to be taken to arrive at the final time. Fewer time steps gives a significant savings in terms of total operations performed.

9 WOFD in Higher Dimensions

The effectiveness of WOFD has been illustrated in 1 dimension. The natural follow-up question would concern the effectiveness of WOFD in higher dimensions. At this point let us break WOFD into two parts: the first part is the grid definition, and the second part is the differencing on this new grid.

Grid definition falls within the realm of local spectral analysis. That is, one is interested in the spectrum locally. Local high frequency data requires a grid density sufficient to resolve the highest frequency, whereas local low frequency data can be resolved with a relatively coarse grid. The wavelet structure provides a very convenient mechanism to perform this nested group of short-time Fourier transforms. For higher dimensions, say 2 dimensions, one need only choose a coordinate system for the space and simply perform the wavelet filtering throughout all of the data and parallel to each axis. The most common question at this point concerns the effectiveness of this method of grid selection when the data is composed of structure which is at a 45 degree angle to the axes. The simple answer is that it works well since all structures within the data can be projected onto the orthogonal coordinate system which spans the space. If, however, one is not satisfied with the grid given in this situation then the parameter which adjusts the sensitivity of grid selection can be adjusted. With this sensitivity adjustment, one will always find a suitable grid. Included here are three sets of data which will illustrate the effective grid selection. The first function is a discontinuity at a 45 degree angle to the axes, see Figure (8), and the grid chosen for this data, see (9). The second set of data corresponds, roughly, to the inner and outer flow near a boundary,

$$f(x,y) = 1 - e^{\frac{-y^2}{x}}, \tag{49}$$

see Figure (10), and the grid chosen in this case, see Figure (11). The third set of data is numerically-generated pressure from a turbulent flow. A contour diagram is given in Figure (12) and the grid chosen is given in Figure (13). In all cases the wavelet is

the D_4 wavelet.

Differencing on a grid chosen by WOFD will depend on the application. The D_4 is the optimal wavelet if one is using a central optimal 4th-order finite difference method in the sense of Section 3. But one is not confined to matching the wavelet precisely to the differencing method. The only recommendation is that the stencil of the finite difference method be of roughly the same size as the length of the wavelet filters. This will insure that in the grid selection one is not filtering data which is too far outside of the support of the differencing stencil. Also, depending on the application one might need a reliable mechanism for choosing locally rectangular grids. In this case one would choose the finest grid produced by WOFD in each rectangular region.

One other issue is the use of WOFD to refine beyond the 'finest grid'. For the 1 dimension case considered in this paper there was always an underlying finest grid. This was a theoretical convenience and not a necessity or even a desirable feature. Based on the energy or magnitude of the smallest scale wavelet coefficients one can refine to any desirable grid density, see [10].

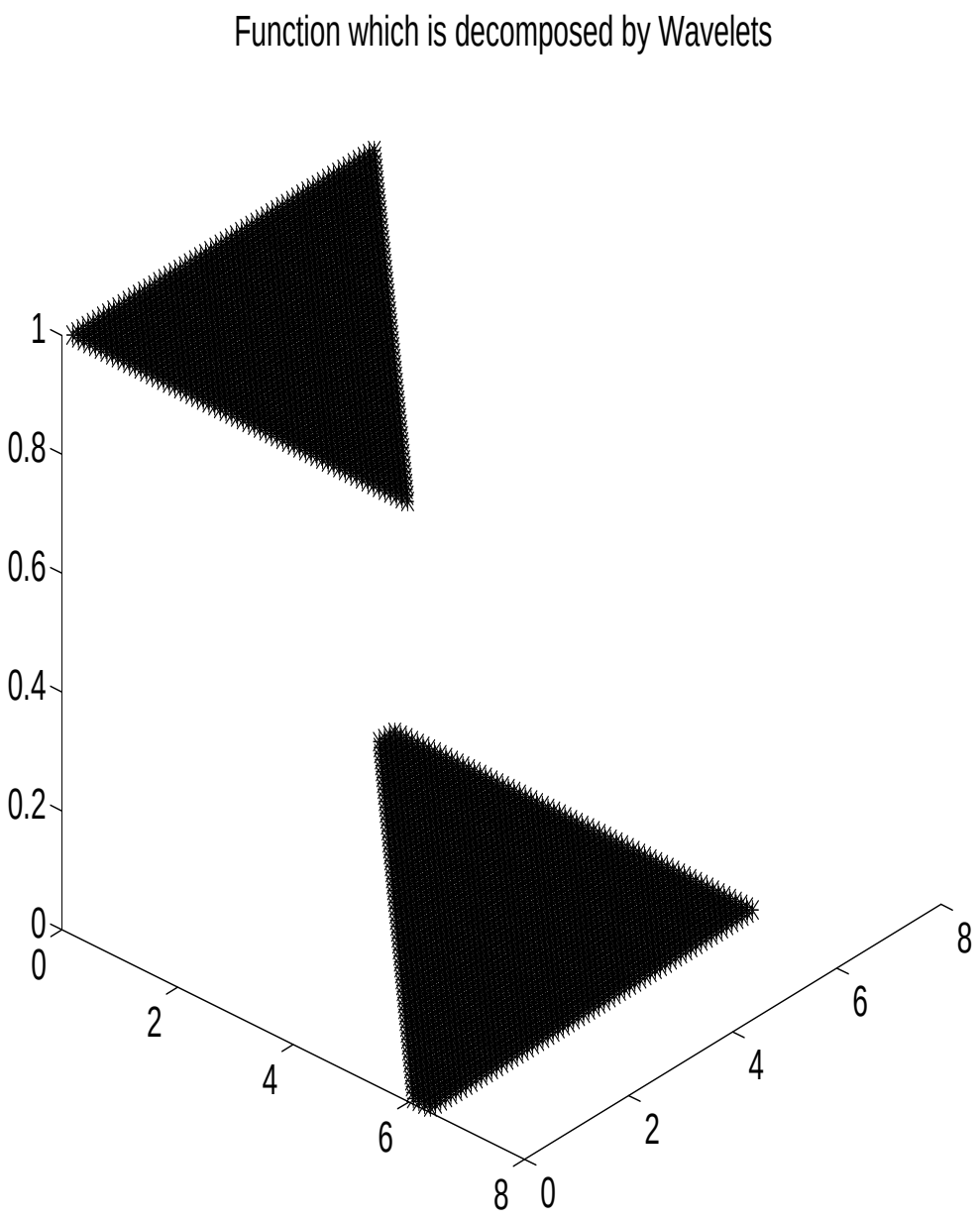

Figure 8: A discontinuity at a 45 degree angle to the axes.

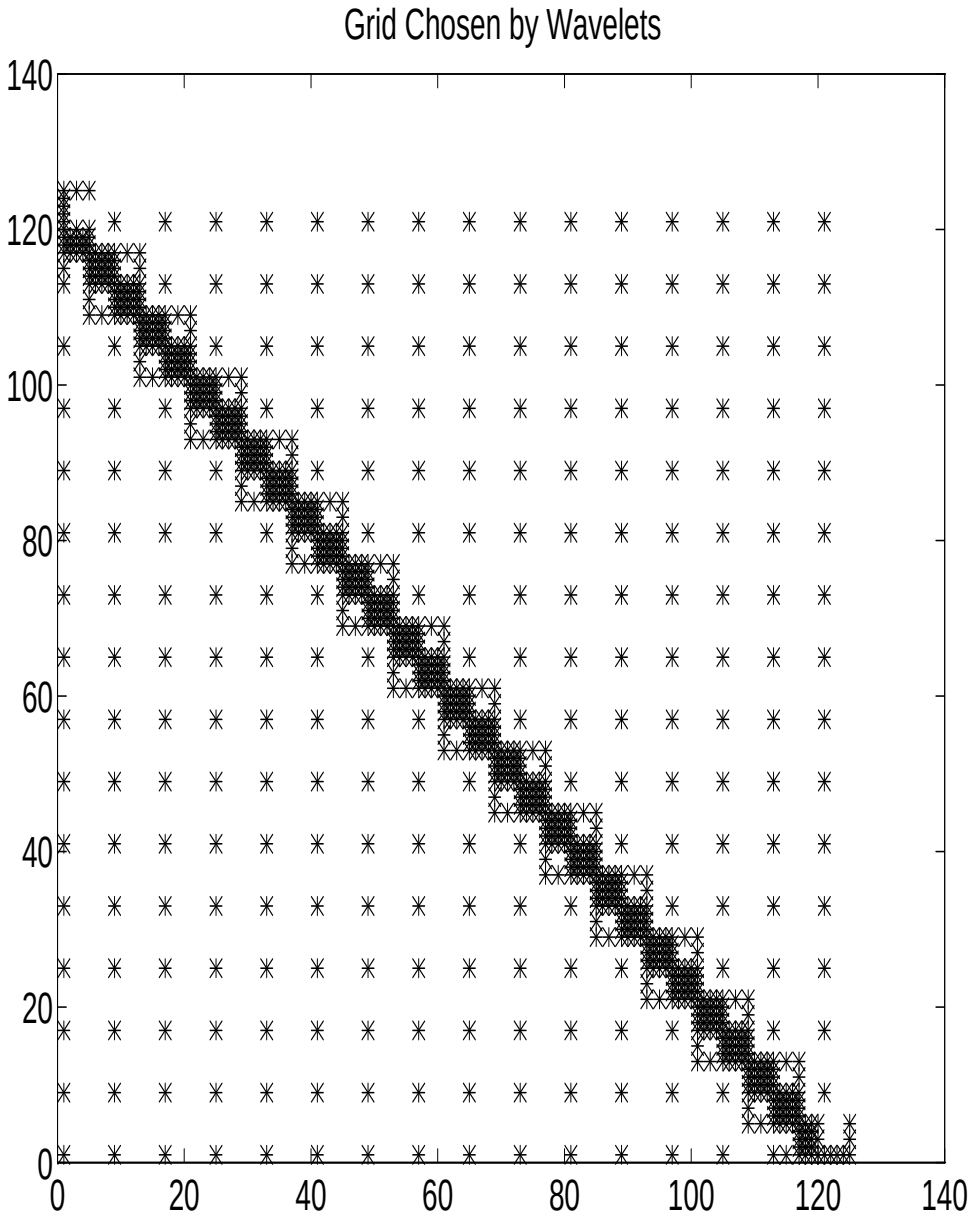

Figure 9: The grid chosen by the wavelet decomposition for a discontinuity at a 45 degree angle to the axes.

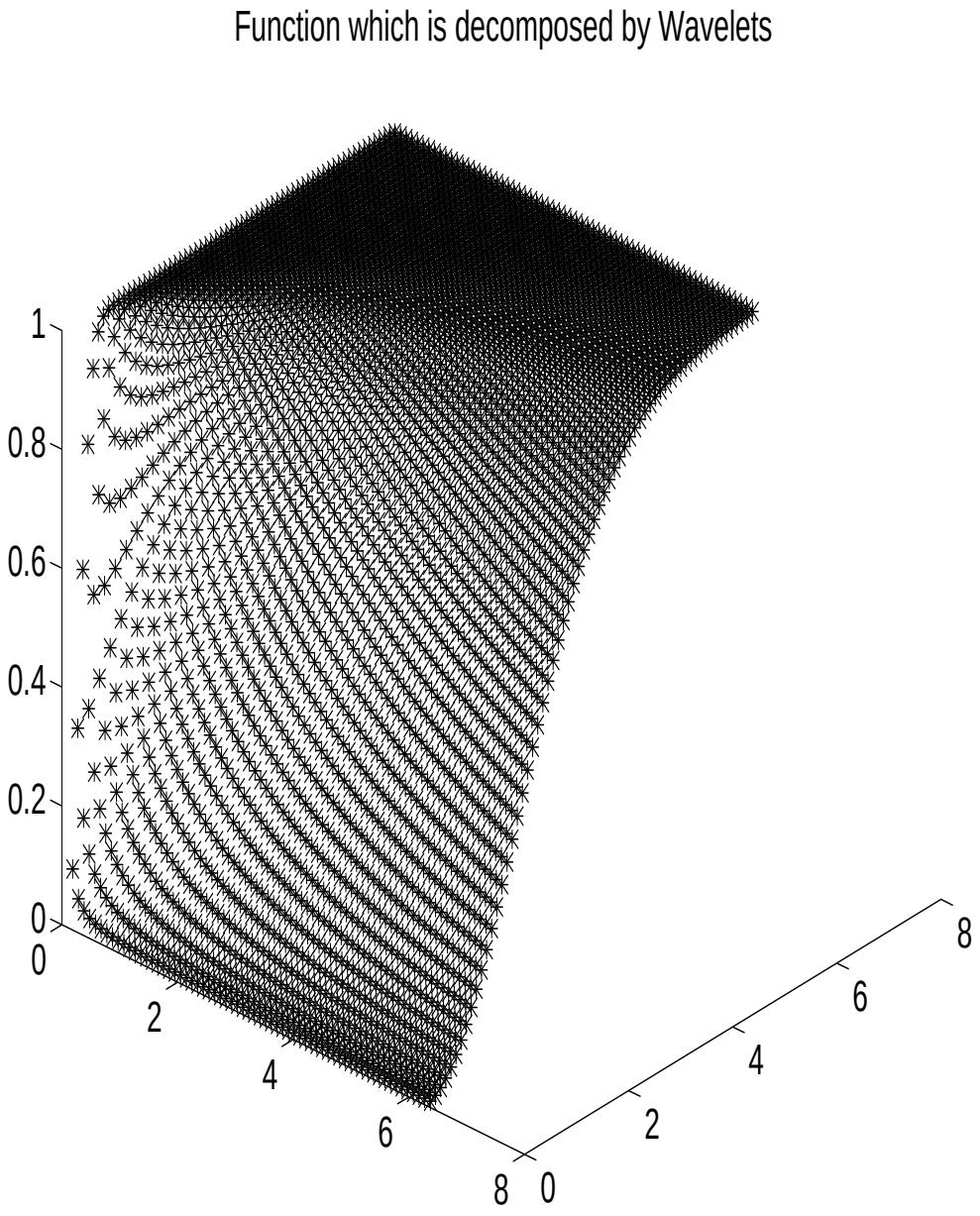

Figure 10: A snapshot of the inner and outer flow near a boundary.

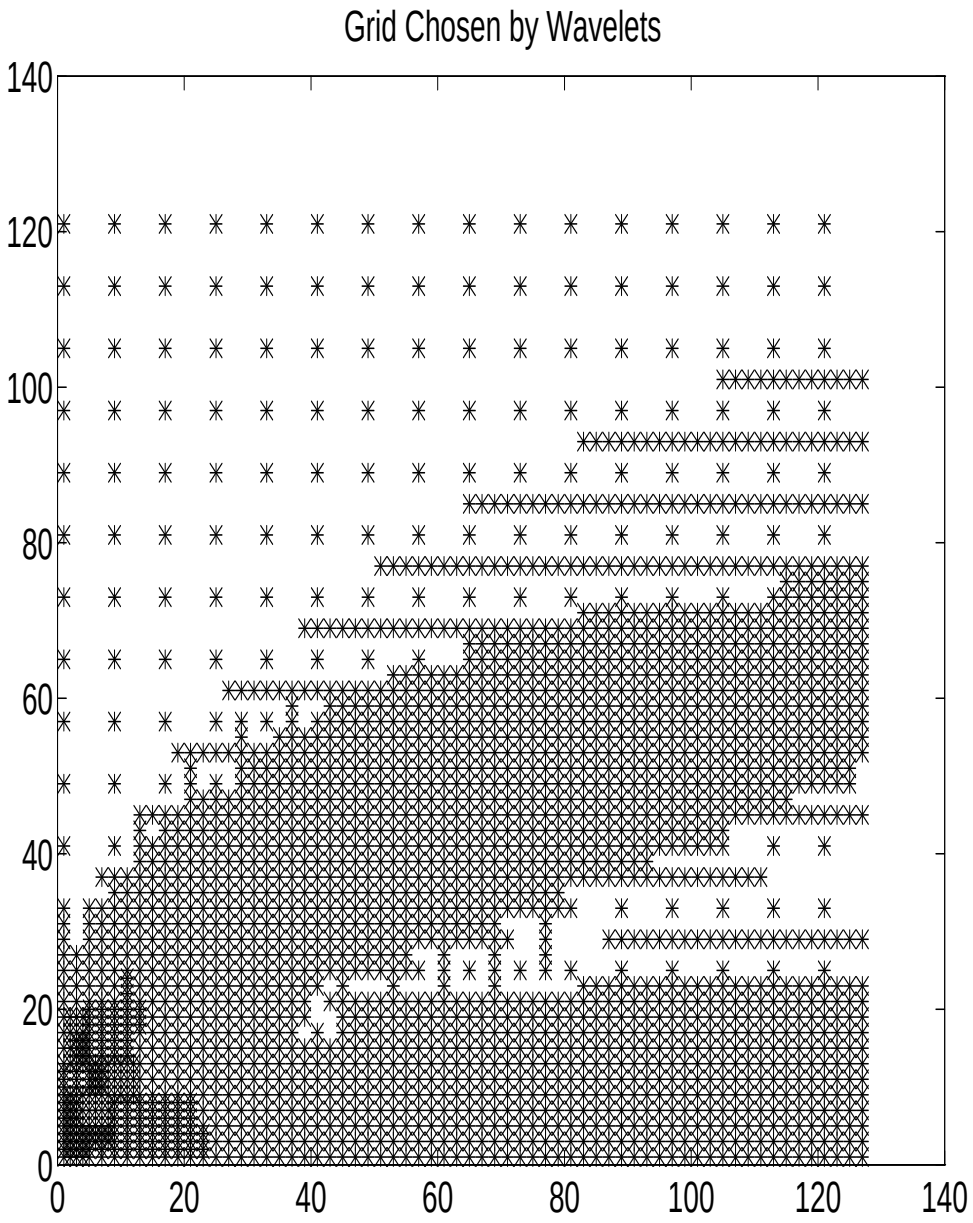

Figure 11: The grid chosen by the wavelet decomposition of the flow near a boundary.

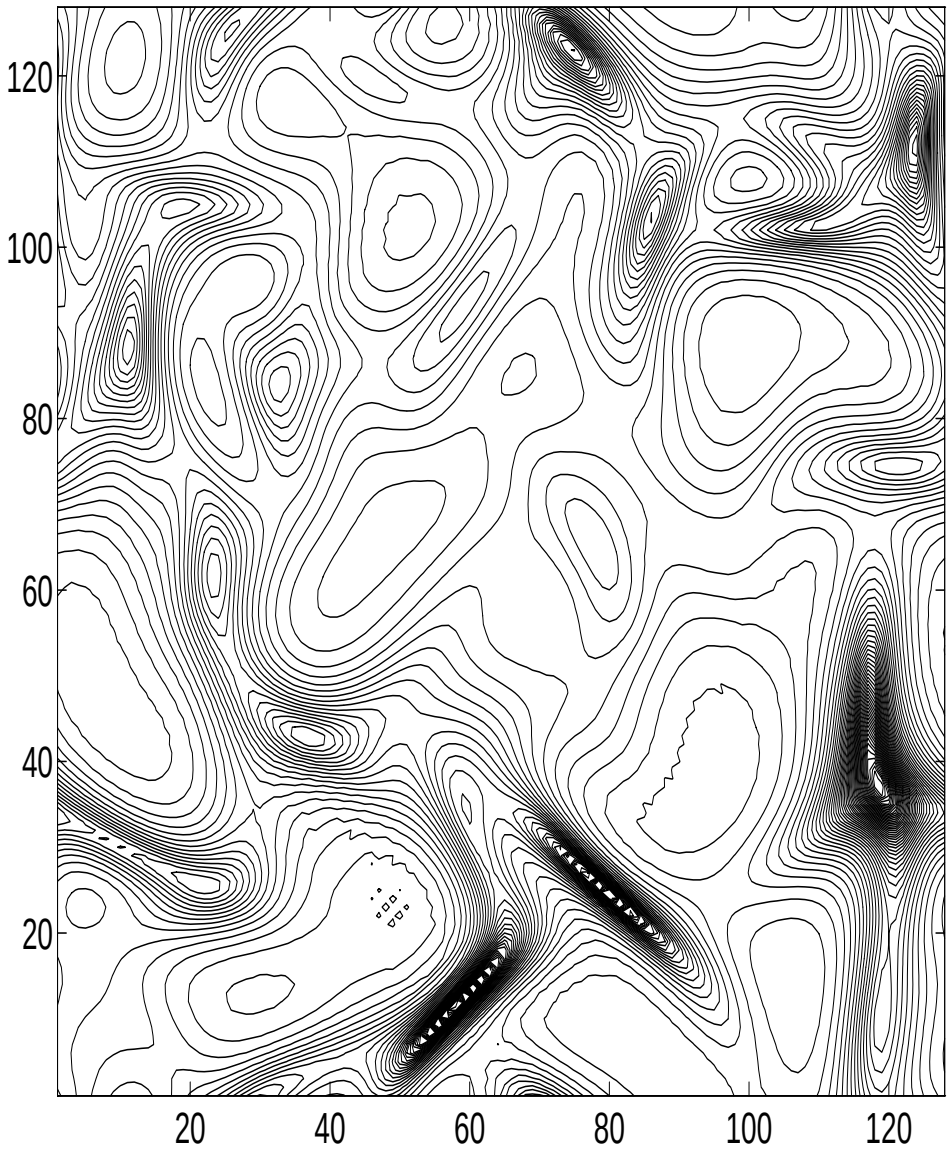

Figure 12: A contour plot of pressure from turbulent data.

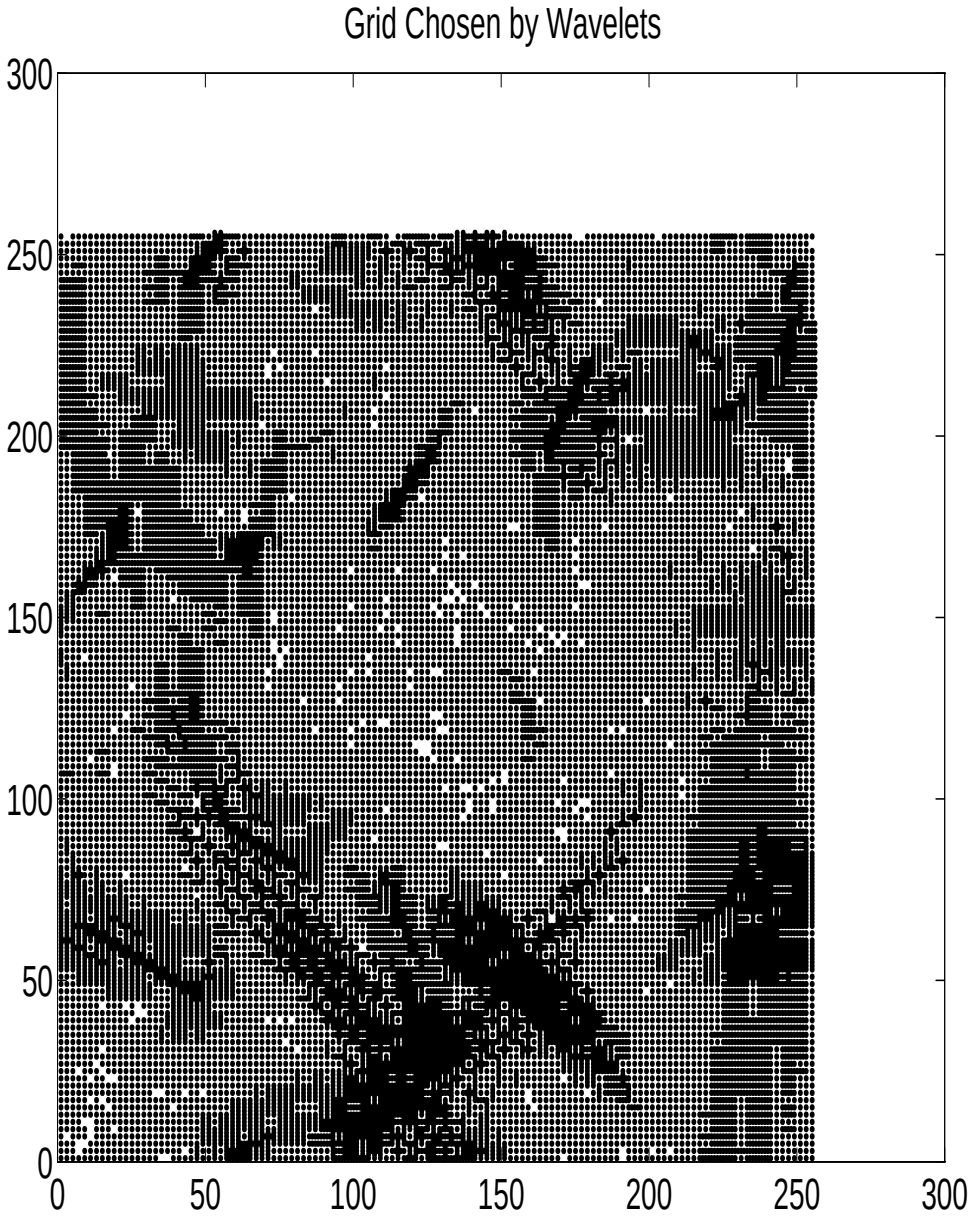

Figure 13: The grid chosen by the wavelet decomposition of the pressure from turbulent data.

10 Conclusion

The WOFD method is essentially the same as a Daubechies-based wavelet method. However, two serious problems encountered with wavelet numerical methods are overcome. That is, boundary conditions can now be imposed in exactly the same manner they are imposed for finite difference methods. Furthermore, there is no longer a problem with nonlinear terms since we are now working with the point values of the function in the physical space. WOFD will approximate a conservative numerical method as long as one is working with a conservative numerical method on the finest uniform grid. This approximation can be as fine as the user wishes.

In this paper the WOFD method has been explored for the case of a 5-point 4-th order finite difference operator on an arbitrary grid. We can, however, use this method with higher order schemes by using the filters associated with the higher order Daubechies wavelets to define the grid. The only suggestion is that the stencil size of the numerical scheme be of roughly the same size as the length of the filters.

Higher dimensional applications of WOFD remain to be explored. As mentioned, WOFD can be broken into two parts, the grid selection and the differencing. The grid selection step requires the wavelet analysis to detect the local oscillation content of the data. Based on this grid, one can choose a number of ways to apply finite differencing. It has been shown here that grid selection is quite effective for 'difficult' data sets in two dimensions. Future research will combine this 2-dimensional grid selection with appropriate differencing.

Acknowledgements

I would like to acknowledge the guidance of my thesis advisor Professor David Gottlieb at Brown University who asked the questions that led to this work.

References

[1] G. Beylkin, "On the Representation of Operators in Bases of Compactly Supported Wavelets", SIAM J. Num. Anal., 29(6): 1716-1740, 1992.

[2] M. Carpenter, D. Gottlieb, and S. Abarbanel, "The Stability of Numerical Boundary Treatments for Compact High-Order Finte-Difference Schemes", ICASE Report No. 91-71, 1991.

[3] A. Cohen, I. Daubechies, and P. Vial, "Wavelets on the interval and fast wavelet transforms", Submitted to Applied and Computational Harmonic Analysis, 1993.

[4] I. Daubechies, "Orthonormal Basis of Compactly Supported Wavelets", Comm. Pure Appl. Math., **41** (1988), pp. 909-996.

[5] Esteban, D. and Garland, C., "Applications of Quadrature Mirror Filters to Split Band Voice Coding Schemes," Proc. Int. Conf. Acoustic Speech and Signal Proc., May 1977.

[6] D. Gottlieb, B. Gustafsson, P. Olsson, and B. Strand, "On the Superconvergence of Galerkin Methods for Hyperbolic IBVP". Submitted to SIAM J. Sci. Comp., 1993.

[7] L. Jameson, "On The Wavelet Based Differentiation Matrix", Journal of Scientific Computing, September 1993 and ICASE Report No. 93-95, NASA CR-191583.

[8] L. Jameson, "On The Spline-Based Wavelet Differentiation Matrix", Submitted to SIAM J. Num. Anal. and ICASE Report No. 93-80, NASA CR-191557.

[9] L. Jameson, "On The Differentiation Matrix for Daubechies-Based Wavelets on an Interval", Submitted to SIAM J. Sci. Comp. and ICASE Report No. 93-94, NASA CR-191582.

[10] J. Liandrat and P. Tchamitchian, "Resolution of the 1D Regularized Burgers Equation using a Spatial Wavelet Approximation Algorithm and Numerical Results", ICASE Report No. 90-83 (1990).